**UNIVERSIDADE FEDERAL DE MINAS GERAIS**

**TRABALHO PRÁTICO: MÁQUINA DE BUSCA**

**DISCIPLINA:PDS2**

**Bruno**

**José Raimundo de Castro Filho**

**Leonardo**

**Belo Horizonte, Junho 2019**

**Introdução**

Uma máquina de busca fornece suporte para a procura de informações armazenadas em um banco de dados. Quando realizamos uma consulta é criada previamente uma lista de ocorrências para podermos cruzar os dados e obter um resultado satisfatório que é saber em quais documentos estão as palavras que procuramos ou quais os possíveis documentos.

Para termos uma idéia de como uma máquina de busca eficiente é importante podemos citar como o Google dominou o mercado de buscadores na internet com uma idéia relativamente simples e similar ao nosso trabalho: A quantidade de vezes que uma página da web era listada em outro site indicaria a sua relevância para os usuários foi nesse momento que surgiu uma das métricas mais utilizadas por diversos buscadores atuais o PageRank. Antes de tal método as buscas eram bem menos eficientes mostrando a importância do tema no mundo atual.

Implementação

Para construir a máquina de busca do presente trabalho criamos três classes,sendo elas: Leitura, Indice e Ranking.

Para isso criamos as classes Leitura, Indice e Ranking. O Fluxograma abaixo ilustra como o trabalho foi conduzido**.**

**Abaixo segue a explicação da implementação:**

Ler os Arquivos

Classes utilizadas (Leitura)

Função – ReadFile

**Para a leitura dos arquivos criamos a função ReadFile que recebe como parâmetro a quantidade de arquivos que será lida. Optamos por criar nosso programa todo embutido nessa função para que a tudo fosse processada a medida da leitura.**

Manipulação das Palavras

Classes utilizadas (leitura, Indice)

Função – TiraCaracter e TudoMinusculo

**Para remover os caracteres da palavra utilizamos a função TiraCaracter, para colocar todas as letras da palavra minúsculo a função TudoMinusculo, ambas recebem como parâmetro uma string que no caso é a palavra que acabou de ser lida do arquivo**

Armazenamento das Palavras lidas

Classes utilizadas(leitura,Indice)

**As palavras de um Arquivo eram inseridas em um list de string wordFile e ao terminar de ler o arquivo essas palavras eram armazenadas em um vector de strings PalavrasDocs**

**Conclusão**